

2D 애니메이션에서 모션 및 본의 활용에 대한 연구

한성호*

I. 서론
1. 연구 목적
2. 연구방법 및 범위
II. 기술동향
1. 모션캡처 기술동향
2. 3D의 2D화 기술동향
3. 2D의 3D화 기술동향
III. 분석 및 평가
1. 분석
2. 평가
IV. 결론 및 향후 연구
참고문헌

I. 서론

1. 연구 목적

모션캡처(Motion capture)는 어떤 실제물체의 움직임을 수치적 데이터로 저장하였다가 컴퓨터로 만든 가상의 물체에 모션 데이터를 넘겨주는 과정을 말한다. 흔히, ‘모션캡처’¹⁾라 함은 실제사람의 움직임을 데이터화하여 가상의 캐릭터(3D Modeler)에 입히는 것으로 알려져 있다. 이러한 경우에는 사람의 관절에 ‘특수마커(센서)’를 부착시키고 마커들의 위치, 회전 데이터를 특수 장치에 의해 실시간으로 인식시켜 ‘모션데이터셋(motion data set)’ 혹은 ‘모

* 동국대학교 영상정보통신대학원 멀티미디어학과 박사과정.

1) <http://web.mit.edu/comm-forum/papers/furniss.html>.

선커브(motion curve)'를 만들게 된다.

키 프레임 애니메이션(Key Frame Animation)과 트위닝(Tweening) 및 시뮬레이션(Simulation)과 같은 기존 애니메이션 기술에 비해 '모션캡처'의 가장 큰 장점은 실시간 애니메이션 영상과 자연스럽게 보이는 높은 질(Quality)의 영상을 제공할 수 있는 장점을 지니고 있다. 1980년대 이후로 꾸준히 연구되고 발전되어오고 있지만 현재 모션캡처 기술이 3D의 전유물처럼 활용되고 있는 것이 현실이다. 그러나 3D애니메이션에서 활용되는 모션캡처(Motion Capture)나 본(Bone)을 통한 캐릭터 애니메이션을 2D 애니메이션 제작에 적용시킬 수 있다면 2D 애니메이션 제작 방법상에 있어서 기념비적인 획을 그을 수 있을 것이다. 3D 애니메이션 제작에서 활용되는 모션캡처를 2D에 적용시키기 위해서 선행되어야 할 과제가 바로 2D 그래픽에 본(Bone)의 적용 문제이다. 본 논문에서는 이러한 점을 중심으로 연구하였다.

2. 연구방법 및 범위

2D 애니메이션에 모션캡처의 기술을 활용할 수 있게 하기 위해서는 3D를 2D화시키든지 아니면 2D를 3D화시켜야 한다. 3D를 2D화시키는 작업 즉 카툰렌더링과 같이 3D로 만든 애니메이션을 렌더링 시에만 2D로 렌더링하는 경우에는 오랜 시간동안 학계나 업계에서 매우 활발하게 연구 및 개발이 진행되어 왔고 지금은 특별한 일이 아닌 듯이 보편화 되어있으나 아직까지 3D의 느낌이 너무 강해 진정한 의미에서의 2D 애니메이션으로 보기 어렵다고 생각한다. 반면에 2D를 3D화시키는 작업을 통해 모션캡처를 활용한 경우는 국내외 논문이나 상업화²⁾된 예가 전혀 없다.

2D를 3D화시킨다는 말은 2D에 3D에서 사용하는 본(Bone) 또는 스켈레톤(Skeleton)을 형성하여 2D이미지 객체와 본을 바인딩(Binding)시킨 후 모션 캡처 데이터를 본에 적용하여 2D 애니메이션을 제작하는 방법을 말한다. 본 논문의 연구방법 및 범위는 후자를 택하여 II. 기술동향에서 모션캡처 및 애니메이션의 기술동향을 정리하고 III. 분석 및 평가에서 2D의 3D화에 대한 제작 원리를 제시하며 IV. 결론 및 향후 연구에서는 본 연구의 의의 및 한계점을 알아보고 향후에 그 한계를 극복해나갈 방향을 제시하고자 한다.

2) 뒤에서 언급될 로스트마블사의 모호(Moho)는 상업화에 성공했으나 2D에서 본만 적용가능할 뿐 모션 적용이 전혀 안되고 2D에 모션이 적용되는 애니문의 모션플래쉬(Motion Flash)는 아직 상업화가 되지 않았다.

II. 기술동향

1. 모션캡처 기술동향

모션캡처란 인체가 동작하는 움직임을 즉시 또는 약간의 시간 차이를 두고 기록하는 작업을 말한다. 기록된 정보를 바탕으로 컴퓨터 캐릭터에게 위치 정보를 포함하는 정보를 전달하여 인체의 움직임이 그대로 컴퓨터 캐릭터로 나타날 수 있도록 한다. 모션캡처가 세상에 널리 알려지기 시작한 것은 1970년대 말부터였다.

모션캡처에서의 위치 정보 전달은 우선 두 가지로 나눌 수가 있는데, 사람의 팔이 움직이는 것과 같이 컴퓨터 캐릭터의 팔이 움직이는 직접 위치 정보 전달이 그 첫 번째이고, 두 번째는 인간의 손에서처럼 손가락의 움직임이라든가, 피부색 그리고 감정적인 표현을 나타내는 간접적 정보 전달로 나눌 수 있다.

인간의 동작을 그대로 복제하여 캐릭터를 움직여서 애니메이션을 만드는 것은 기존에 아주 새로운 것은 아니었고 미국의 디즈니 스튜디오에서 인간의 동작을 수작업을 통해 복사하는 로토스코핑(Rotoscoping)³⁾ 작업을 이미 하고 있었고, 1970대 말경에 컴퓨터를 이용해서 기존 로토스코핑의 수작업 방식을 포함한 애니메이션 기법이 생기게 되었다.

비슷한 시기에 뉴욕의 한 컴퓨터 그래픽 실험실에서 알렌(Rebecca Allen)이라는 사람은 반사경을 이용해서 비디오테이프에 녹화된 댄서의 춤사위를 컴퓨터에 그래픽으로 저장하는 작업을 하고 있었다. 빛을 이용해 스크린에 나타난 인간의 동작을 반사경을 이용해 복사를 해서 컴퓨터에 저장을 하여 사용한 것이다.

1980~1983년 즈음에 컴퓨터를 이용해서 인간의 동작을 분석하기 위한 움직임이 학문적으로 시작되었다. 이러한 움직임의 결과가 컴퓨터 그래픽 사회에 영향을 미치게 되었다. 1980년대 초기 Simon Fraser 대학의 Tom Calvert 교수는 통신분압기를 인체의 양쪽 무릎에 부착하고 모션캡처하여 컴퓨터 캐릭터를 움직이는 방법을 처음으로 고안하였다.

1993년 상품화된 모션캡처 시스템이 출현하기 이전에는 복잡한 오브젝트나 동작의 애니

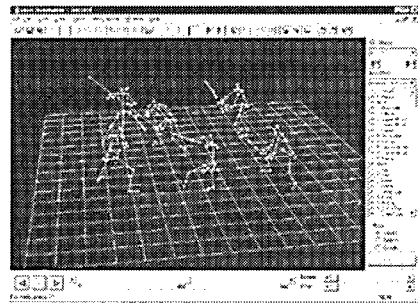
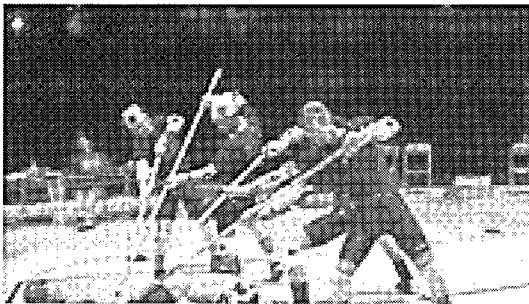
3) 1915년 맥스 플레이셔(Max Fleicher)에 의해 고안된 로토스코프(Rotoscope) 방법은 실제 사람의 움직임을 연속 사진에 담고 이를 그림판에 투영시켜 애니메이터로 하여금 윤곽선을 따라 그림을 그리도록 한다. 이러한 방식을 사용하면 매우 사실적인 움직임을 애니메이션으로 재현할 수 있어 애니메이션 제작에 폭 넓게 활용되었다. 디즈니의 <백설공주>나 <아나스타샤>에서 로토스코프를 이용한 실사 영화와 같은 느낌의 애니메이션 제작하는데 사용되었다.

메이션 제작에 움직임의 주요 장면(Keyframe)을 수작업에 의해 생성하고 이들 사이의 중간 장면을 채우는 방식으로 제작되는 키프레임 애니메이션과 물리법칙 등 일련의 규칙들에 의해 오브젝트의 움직임을 자동 혹은 반자동적으로 생성하는 4)동작제어(Motion Control)기술에 의한 방법을 사용하였다.

그 후, MIT 공과 대학과 New York Institute of Technology에서 광센서를 이용한 모션 캡처 방식이 고안되어 실험을 하였다. 인체에 부착하는 LED로부터 각 센서의 2차원 값을 두 대 이상의 카메라를 통해 얻은 값을 통합하여 3차원 정보를 얻어내는 기술로 발달하게 되었다. 이와 관련하여 조합된 3차원 정보의 렌더링(Rendering)기술이 발달하여 현재까지 이 방법이 고가임에도 불구하고 가장 보편화된 기술이 되었으며 상업성을 띤 작품들에 모션캡처 기술이 응용되기 시작되어 뮤직 비디오 촬영이나, TV 어린이 프로그램 등에서도 응용이 되었다. 매년 시그라프(SIGGRAPH)행사를 통해 모션캡처 기술과 응용 범위의 발달이 소개되고 있는데, 이제는 대학과 같은 학문 연구 기관보다, 상업적 목적을 가진 기업들이 더 앞장서고 있는 형편이다.

따라서, 1993년 이후부터는 Ascension, Polhemus, SuperFluo와 같은 회사들이 상업용 제품을 출시하였고, 앞으로 모션캡처 기술은 다양한 방법으로 발달이 되고 그 응용 범위가 다양해 질것으로 보인다.

다음 <그림 1>은 최초로 모션캡처시 3인 이상을 활용하여 2000년 시그라프(SIGGRAPH) 최우수상을 수상한 일본 게임 <귀무자> 모션 캡처 사진이다.



4) 김용순, 김영수, “3차원 캐릭터 애니메이션 기술동향”, <정보과학회지> 17권 2호, 1999.



<그림 1> 게임 <귀무자>의 모캡장면과 오프닝 동영상 장면

최근 컴퓨터를 이용한 애니메이션 작업의 대부분은 3차원 영상으로의 이동과 이러한 기술적 발전을 바탕으로 현실감 넘치는 가상현실(Virtual Reality)의 구현이라고 할 수 있다. 가상현실 공간이 상상의 속도만큼 넓어지는 만큼 그 공간 내에서 움직이는 캐릭터들도 바쁘게 움직여야 할 것이다. 하지만, 대부분의 캐릭터들은 현실에서 가공되는 인물들이고 그들의 움직임에 대부분의 시간이 투자되는 현실이 되었다.

모션캡처 장치가 없다고 한다면, 가상공간 즉, 게임이나 영화와 같은 가상공간 내의 캐릭터들을 움직이도록 하는데 투자되는 현실의 시간 때문에 가상현실 공간이 발전하는데 커다란 걸림돌이 될 것이다. 따라서, 모션캡처 시스템은 가상현실을 더욱 현실감 있도록 하기 위한 근본적인 툴(Tool)이라고 말할 수 있다.

2. 3D의 2D화 기술동향

3D의 2D화란 실사 지향 및 사실적인 느낌위주의 3D그래픽을 2D애니메이션과 같은 분위기로 렌더링을 해주는 기법을 말하며 흔히 만화처럼 보이게 한다 하여서 카툰렌더링(Cartoon Rendering)이라고 칭하며 이외에 셀 렌더링, 비 실사 렌더링, 도트 렌더링 등으로 부르기도 한다. 일반적으로 카툰렌더링과 셀 렌더링은 같은 의미로 활용되지만, 비 실사렌더링(Non-Photorealistic Rendering, NPR)은 좀 더 큰 범주에 속하는 기법이라고 할 수 있다. 비실사렌더링 분야에는 카툰렌더링 이외에도 연필 스케치, 수채화, 흑백 일러스트레이션, 유채화, 기술도면 렌더링 같은 다양한 양식화된 렌더링 기법들이 포함된다. 카툰렌더링의 가장 기본적인 목표는 만화 같은 느낌의 부여다. 특히 셀 애니메이션에 최대한 근접한 느낌을 주고자 하는 것이 이 기법을 사용하는 목적이다. 이 방식은 기존 2D그래픽의 섬세함과 3D그래픽

의 다양함을 적절히 섞어서 보는 이로 하여금 거부감 없이 받아들일 수 있게 해준다. 즉, 3D가 다양함 측면(다양한 카메라 워, 조명효과, 매핑효과 등)에선 좋지만, 금속성면에서 부담스럽고, 2D가 따뜻함은 주지만 다양함이 부족한 면을 감안, 장점만을 합친 것이다. 카툰렌더링 기법은 3D가 사실적인 표현을 위해 하이라이트, 중간 밝기 그리고 어두운 단계로의 자연스러운 그라데이션을 표현하는 데 비해 가장 밝고 가장 어두운 부분의 색만을 추출하고 거기에 윤곽선을 진하게 표현해 2D애니메이션과 같은 느낌을 제공한다. 중간 톤을 생략해 전체적으로 채도를 높여 밝고 따뜻한 느낌을 준다. 카툰렌더링은 표현해야 하는 색의 단계가 적기 때문에 렌더링 시간을 줄일 수 있으며, 사실성이 떨어지는 애니메이션도 자연스럽게 보일 수 있다는 특징을 갖고 있다.

좀 더 기술적으로 살펴보면, 3D의 2D화는 모델의 외곽선 변(Silhouette edge)을 검출하여 모델의 테두리 선들과 주름선, 그리고 재질을 구분하는 경계선을 그리는 5)선화(inking) 작업과 모델의 재질(material) 속성들과 조명 값들에 기반을 두어서 그 모델에 입힐 텍스처 맵을 만들고 정점의 법선 등에 기반을 두어서 계산된 텍스처 좌표를 이용해서 객체의 면들에 그 텍스처를 입히는 작업 즉 모델의 외곽선 안을 칠하는 6)채색(painting) 작업으로 이루어진다. 선화 과정은 크게 두 부분으로 나뉘는데 첫 번째는 외곽선 변들을 검출하는 것이고, 두 번째는 그들을 이용하여 외곽선을 그리는 것이다. 채색 과정 또한 두 단계로 나눌 수 있는데 하나는 전처리 단계이고, 다른 하나는 실행 시점의 단계이다. 전처리 단계에서는 각 재질마다 재질의 조명과 재질 속성들에 기반을 두어서 텍스처 맵을 만든다. 실행시점 단계에서는 각 정점의 법선과 조명 벡터 사이의 각도에 기반을 두어서 텍스처 좌표들을 생성하고 그것들을 이용해서 표면에 텍스처를 입힌다.

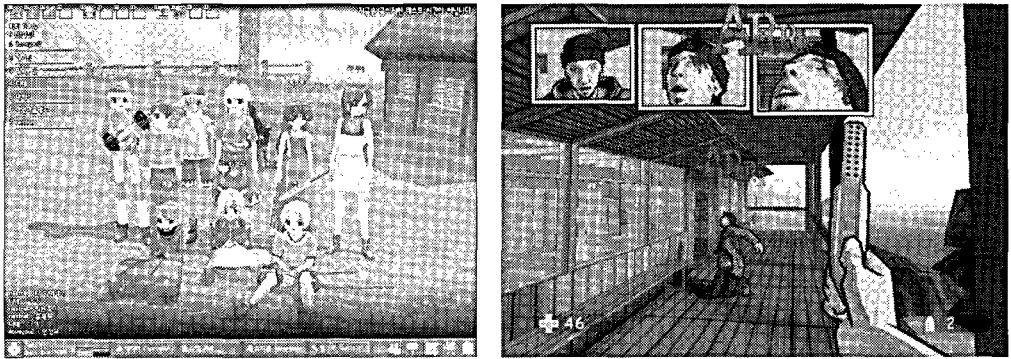
렌더링(Rendering)이란 그림자나 색상과 농도의 변화 등과 같은 3차원 질감을 넣음으로써 컴퓨터 그래픽에 사실감을 추가하는 과정을 말한다. 렌더링의 또다른 형태에는 이미지를 레이트레이싱처럼 객체별로 하는 것이 아니라, 한 번에 한 개의 수직선으로 렌더링하는 주사선 렌더링(ScanLine Rendering)이 있다. 일반적으로 주사선 렌더링이 레이트레이싱에 비해 좋은 결과를 내지는 못하지만, 개별 프레임의 이미지 품질이 그리 중요하지 않은 애니메이션 패키지에서 자주 사용되고 있다. 카툰렌더링이란 이러한 주사선 렌더링 방식을 사용하여 실

5) Carl S. Marshall, "Cartoon Rendering: Real-time Silhouette Edge Detection and Rendering", Game Programming Gems 2, Charles River Media, 2001, pp.437-439.

6) Adam Lake, "Cartoon Rendering Using Texture Mapping and Programmable Vertex Shaders", Game Programming Gems 2, Charles River Media, 2001, pp.444-449.

사 지향 및 사실적인 느낌위주의 3D그래픽을 2D애니메이션과 같은 분위기로 만들어 주는 기법을 말한다.

최근 카툰렌더링 방식을 사용해 출시된 게임을 살펴보면, 온라인게임의 경우 국내 최초 완전 3D 카툰렌더링 온라인게임 <셀 온라인>, 넥슨이 3년간 70억 원을 들여 만든 게임 <마비노기>, 소프트맥스의 첫 온라인게임 <테일즈워버>, 싸이미디어의 <믹스마스터온라인> 등이 있다. 또, PC게임에서는 지오램프의 <장화신은 고양이 토토>, 그리곤엔터테인먼트의 무협PC게임 <천랑열전>, 몽클엔터테인먼트의 <테일즈 오브 윈디랜드>, 손오공의 <하얀마음백구 2>, 가람과바람의 <나르실리온> 등이 있으며 콘솔게임으로는 세계최초로 카툰 렌더링을 적용한 1인칭 슈팅게임(FPS)으로 플레이스테이션 2(PS2)게임 <써틴(XIII)> 등이 있다. 다음 <그림 2>는 넥슨의 마비노기와 써틴(XIII)의 스크린 샷이다.

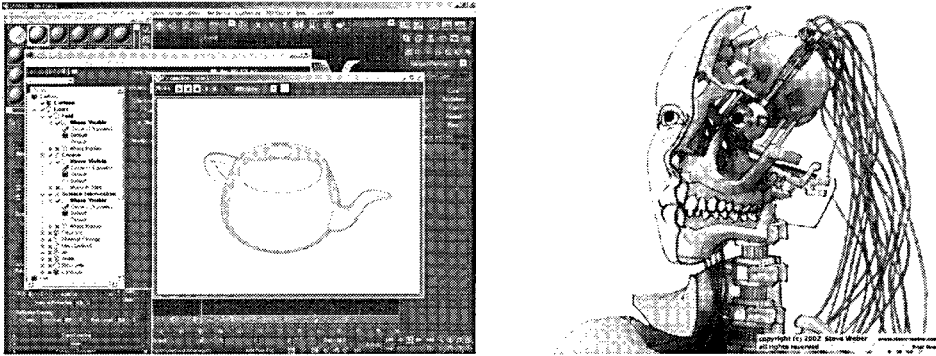


<그림 2> 카툰렌더링을 적용한 마비노기와 써틴(XIII) 게임의 한 장면

이들 게임들의 특징은 동화풍의 분위기로 기존의 10대 청소년만이 아니라, 여성 및 2D 만화에 익숙한 20대에서 컴퓨터의 금속성에 덜 익숙한 유아 층까지 타깃으로 삼고 있다는 점이다. 카툰렌더링 방식이 국내 게이머들의 취향에 부합하는 측면이 크기 때문에 국민 정서상 딱딱한 느낌보다는 부드럽고 따뜻한 느낌을 선호하고, 게임인구의 급속한 증가로 이미지의 다양함에 대한 요구가 크다는 것이다.

카툰렌더링 분야에서 독보적인 자리를 고수하고 있는 David Gould사의 일러스트레이트(Illustate) 프로그램은 셀/카툰 스타일 렌더링 및 벡터 렌더링 기능을 통해 탁월한 카툰 이미지를 창출할 수 있을 뿐만 아니라 다양한 선(Line)스타일을 제공하여 일러스트레이션 스타일의 이미지를 쉽게 창출할 수 있기 때문에 만화영화, TV, 인터넷, 미술 등은 물론 건축, 상업 광고 및 산업 디자인 등의 다양한 분야에서 그 활용도가 매우 높은 프로그램이다.

이와 유사한 프로그램으로 Vecta3D, Comicshop, Cartoon Reyes 등이 있지만 기존 프로그램으로 구현하지 못하였던 렌더링시 선 감추는 기능(Hidden Lines Rendering)과 신속한 재질 변환(Material Convert)기능이 포함된 CEBAS사의 finalToon이 있다. 아래 <그림 3>은 Illustate와 finalToon으로 렌더링한 2D 이미지이다.



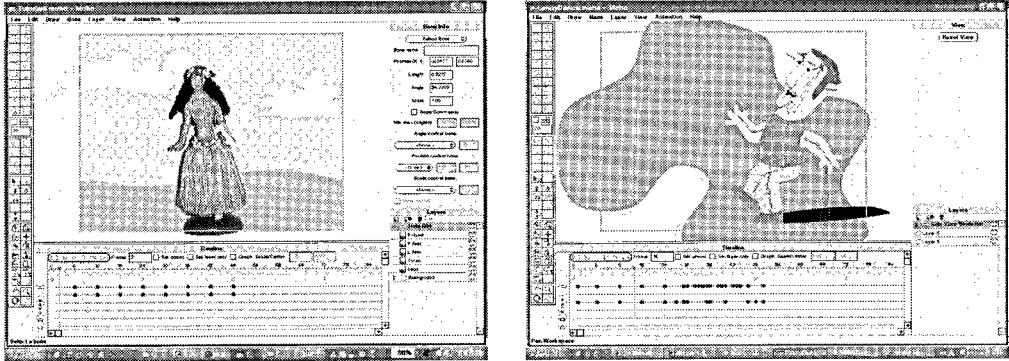
<그림 3> Illustate와 finalToon으로 렌더링한 2D 이미지

3. 2D의 3D화 기술동향

2D의 3D화란 2D에 3D에서 사용하는 본(Bone) 또는 스켈레톤(Skeleton)을 형성하여 2D이미지와 본을 바인딩(Binding)시킨 후 모션 캡처 데이터를 본에 적용시켜 2D 애니메이션을 제작하는 방법을 말한다. 이러한 기술에 대한 논문이나 상용프로그램이 전무한 실정이지만 다소 유사한 상용 프로그램으로 외국 업체의 로스트 마블(Lost Marble)사의 모호(Moho)라는 프로그램에서 업계 최초로 2D에 본(Bone)을 적용시켜 애니메이션을 제작할 수 있게 3D의 개념을 2D에 도입시켰다. 모호에서는 모든 형태의 IK⁷⁾가 적용된 본 제작이 가능하며 본과 이미지 객체 간의 바인딩이 완벽하게 지원된다. 그러나 아쉽게도 모호는 3D에서처럼 본에 모션캡처 데이터를 적용시키는 단계는 지원이 안 되고 있다. 정확히 말하면 모션캡처 데

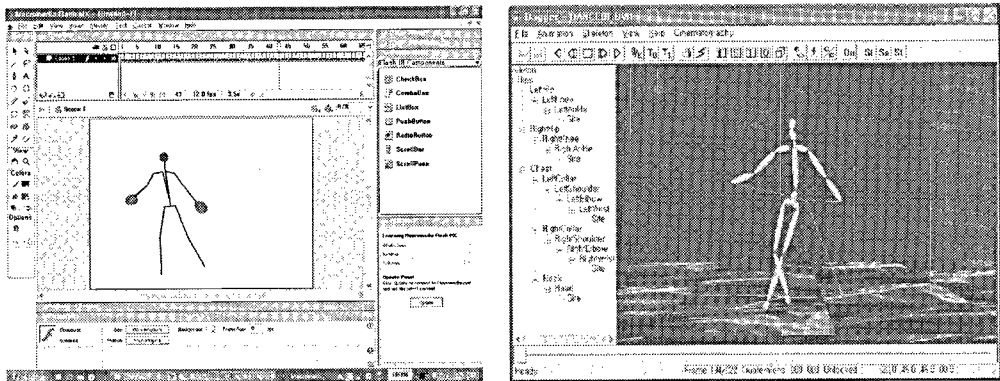
7) IK는 Inverse Kenematic로 캐릭터에 역방향 운동학 시스템이 적용된 개념이다. 모션캡처 과정의 결과는 애니메이션의 다양한 모습을 조절할 수 있는 몇 가지 트랙(Track)의 동작들이다. 각 트랙은 한 채널(Channel)에 할당되고, 보통 각 채널은 3차원 장면에서 물체의 움직임을 조절한다. 채널에서의 동작 데이터는 대개 함수 곡선의 형태로 대부분의 애니메이션 프로그램에 의해 주어진다. 일단 캡처된 동작들의 모든 데이터들이 각각의 채널에 운반되면 그 데이터는 각기 다른 관절(Joint)에 연결될 수 있다. 그 시점에서 골격에 역방향 운동학 시스템을 적용시키는 것이 가능하다.

이터를 전혀 사용할 수 없다. 왜냐하면 본을 모션캡처 데이터에 적용시키기 위해서는 뒤에서 언급할 구조화(Hierarchy)된 본을 사용해야 되는데 이 프로그램은 전혀 그렇지 못하다. 다음 <그림 4>는 모호에서 다양한 본을 제작하는 장면과 본과 객체에 바인딩을 적용시켜 애니메이션을 제작하는 장면이다.



<그림 4> 모호(Moho)에서 본(Bone) 제작 및 바인딩 적용시킨 애니메이션 스크린 샷

국내업계 중에 애니문⁸⁾이라는 회사에서 만든 모션플래쉬(Motion Flash)는 모호와는 구별되게 본에 모션을 적용시켜 2D의 3D화에 어느 정도는 근접하였으나 모호처럼 자체적으로 본을 생성하지 못하는 또 다른 한계점을 갖고 있다. 다음 <그림 5>은 모션플래쉬(Motion Flash)에서 모션을 불러들여 모션을 본에 적용(왼쪽 그림)시킨 후에 플래쉬에서 본을 불러들여 애니메이션을 제작(오른쪽 그림)하는 장면이다.



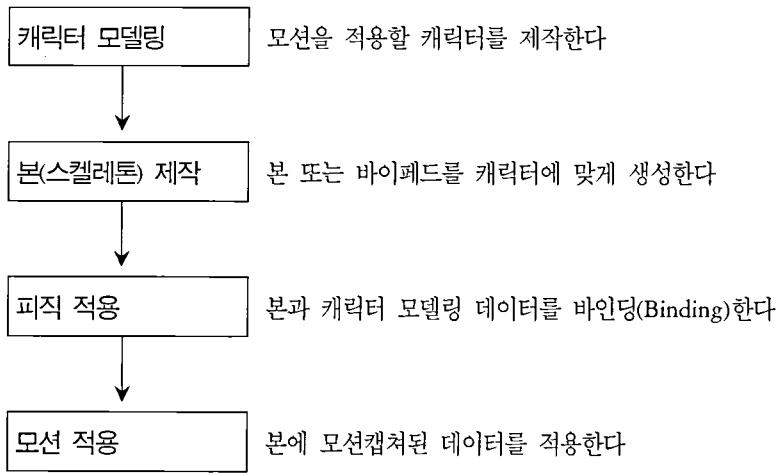
<그림 5> 모션플래쉬에서 모션을 본에 적용시킨 후 플래쉬에서 본을 불러들인 장면

8) 애니문은 애니메이션 콘텐츠 개발관련 회사로 모션플래쉬는 아직 개발단계에 있으며 상용화된 것은 아니다.

III. 분석 및 평가

1. 분석

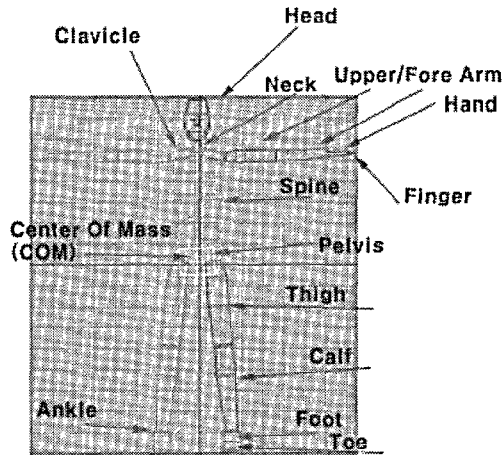
2D의 3D화를 하기 위해서는 3D에서 모션을 적용하는 기법을 2D에 적용시키면 가능할 것이다. 3D에서 모션 적용의 파이프라인은 3DS MAX를 비유로 <그림 6>과 같다.



<그림 6> 3D에서 모션 적용의 파이프라인

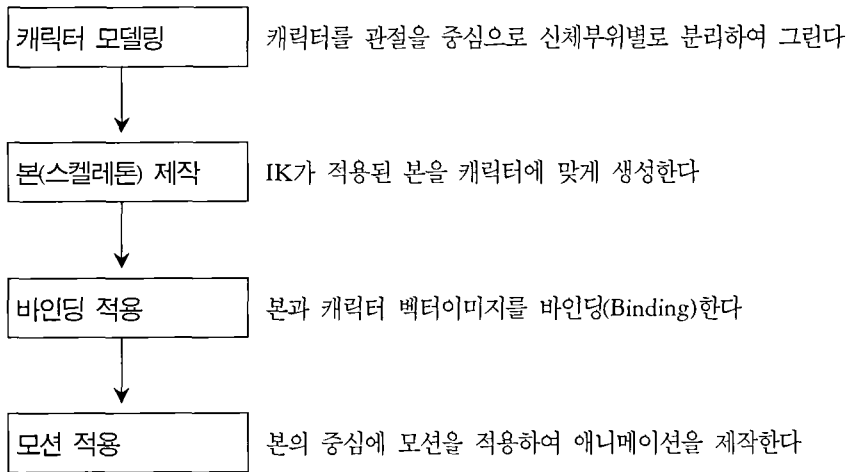
3D의 모션적용 파이프라인을 활용하여 2D에 적용하기 위해서 가장 중요한 2가지는 2D 이미지는 점 방식의 픽셀(Pixel) 이미지가 아닌 선 방식의 벡터(Vector)이미지이어야 한다. 왜냐하면 이미지는 본에 의해서 영향을 받게 되는데 픽셀방식일 경우에 이미지 왜곡현상이 심하게 나타나게 되기 때문이다. 한마디로 이동변형, 회전변형, 크기변형 중에 크기변형⁹⁾에서 문제가 발생하게 된다는 말이다. 다음으로는 모션캡처 데이터를 2D에 적용을 시키기 위해서는 2D의 본의 구조가 3D에서 사용되는 본의 구조에 맞게 설계되어야 한다. 일반적으로 3D의 본의 구조는 맥스를 예를 들어 다음과 같다.

9) 변형(Transform)에는 이동, 회전, 크기 변형이라는 3가지 유형이 있는데 2D의 3D화에서 주의할 것은 본은 크기변형이 일어나지 않지만 본에 바인딩되어 움직이게 되는 2D 이미지는 크기 변형이 일어날 수 있다는 것이다. 따라서 본 적용 캐릭터의 왜곡 현상을 방지하기 위해서 이미지 객체는 벡터(Vector) 방식이어야 한다.



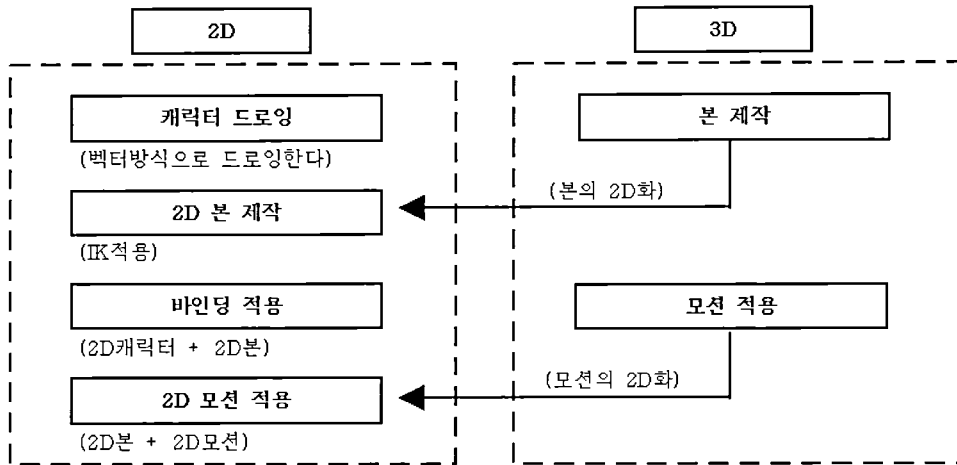
<그림 7> 3DS Max에서 본(Biped)의 구조

상기 2가지 사항을 유의하여 2D의 모션적용 파이프라인을 적용하면 <그림 8>과 같다.



<그림 8> 2D의 모션적용 파이프라인

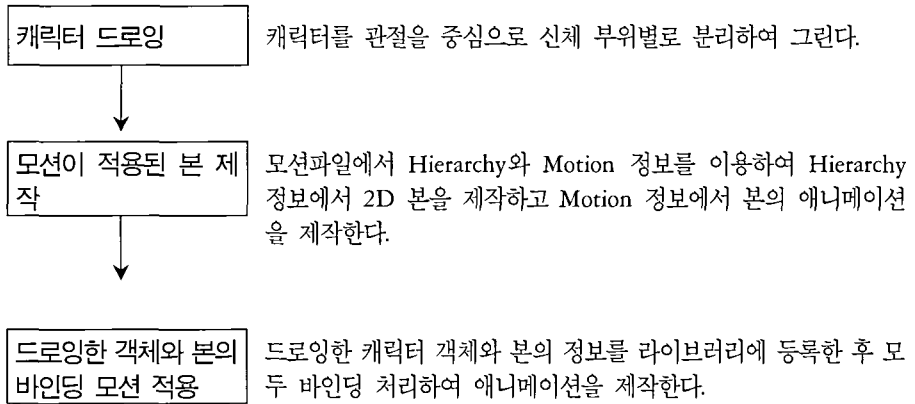
2D의 모션적용 파이프라인에서 캐릭터 드로잉은 2D 작업이고, 본 제작은 3D, 바인딩 적용은 2D로 캐릭터 드로잉한 것과 3D의 본을 x, y 좌표로 2D화한 것 이 2가지를 연동하는 것이다. 마지막으로 모션 적용은 3D의 모션 좌표를 2D화하여 본의 중심점(Root Node)에 적용하면 된다. 이러한 적용과정을 2D와 3D로 분류하여 표를 만들면 <그림 9>와 같다.



<그림 9> 2D와 3D로 분류하여 2D의 3D화 모션적용 파이프라인

2D의 3D화에서 가장 핵심적인 3가지 절차는 본의 제작 둘째, 그리고 본과 객체를 묶는 바인딩 적용, 마지막으로 모션의 적용이다. 상기(上記) 방식의 파이프라인이 가장 이상적인 방법이지만 이와 같은 방식은 모션을 수(手)작업으로 직접 만들지 않고 모션캡처 받은 모션데이터를 적용시켜 애니메이션을 제작하는 경우에는 2D 본 제작 영역에 많은 제약이 주어지게 된다. 모션 캡처 받을 당시의 본 구조로 2D 본을 제작할 경우에는 문제가 되지 않지만 신체의 등선이 서로 다르거나 예를 들어 머리나 팔 등 신체의 특정 부위가 기형적 본 구조를 취하는 모션일 경우에는 마지막 단계인 모션 적용에서 왜곡된 애니메이션이 산출될 경우가 발생되기 때문이다.

이러한 문제점을 극복하기 위해서는 앞서 제시한 2D의 3D화 모션 적용 파이프라인에서 2단계인 본 제작 처리과정과 4단계인 모션 적용 처리단계를 함께 수행되게 하는 방안을 모색해 볼 수 있다. 물론 이렇게 할 경우에는 다른 문제가 발생할 수 있다. 즉 모션 적용을 바인딩 보다 앞서 처리하게 됨에 따라 바인딩 후에 다른 모션 적용이 어렵다는 또 다른 문제가 야기된다. 그렇지만 모션 편집 및 합성에 대한 프로그램은 이미 상용화 되어 있는 많은 소프트웨어가 있으므로 그 문제에 대한 해결 방안은 우회적으로 처리하는데 어려움이 없다. 다음 <그림 10>은 2단계와 4단계를 함께 처리하도록 하는 새로운 방식의 모션 적용 파이프라인이다.



<그림 10> 2D의 3D화 모션적용 파이프라인

2D에서 본 제작 방법은 IK가 적용된 본 객체를 직접 드로잉해서 생성하는 방법과 모션 데이터 안에 기술되어 있는 본 정보를 그대로 구현하는 2가지 방법이 있다. 전자는 다양한 본을 제작할 수 있는 장점이 있는 반면에 모션 적용에 어려움이 따르고 후자는 모션데이터에서 제공하는 본 정보를 따로 불러와서 본을 구현시키는 원리로서 모션 적용은 쉬우나 전자와 같이 다양한 본 제작에는 어렵다는 단점이 있다. <그림 8>이나 <그림 9>와 같은 4단계 파이프라인에서라면 전자가 적용되겠지만 본(Bone)제작과 모션적용이 목적인 본 논문은 <그림 10>의 3단계 파이프라인으로 연구하고자 한다.

모션캡처 장비에서 캡처된 BVH¹⁰⁾ 모션파일 포맷(Format)의 구조를 살펴보면 크게 관절(Joint) 간의 계층구조(Hierarchy)와 상대적 위치(OFFSET), 그리고 자유도(Degree of Freedom, DOF)를 Tree 구조로 나타내는 부분과 각 관절(Joint)의 자유도(DOF)에 따른 프레임(Frame)별 변수들의 변화(Motion)를 나타내는 부분으로 나누어진다. BVH 모션 파일은 텍스트 파일로 정의 되어 있어 일반적인 에디터로도 그 내용을 확인해 볼 수 있다. 다음은 BVH 파일을 텍스트 뷰어로 열어본 내용이다.

HIERARCHY

```

ROOT Hips
{
    OFFSET 0.00    0.00    0.00
  
```

10) BVH 파일 포맷은 모션캡처 서비스 전문회사인 바이오비전(Biovision)에서 개발한 파일 포맷으로 BioVision Hierarchical data를 의미하고 그 성능이 매우 우수하여 널리 사용되는 모션 포맷이다.

CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation

JOINT Chest

{

 OFFSET -0.160000 5.210000 0.160000

 CHANNELS 3 Zrotation Xrotation Yrotation

 JOINT Neck

 {

 OFFSET 0.000000 20.799999 0.000000

 CHANNELS 3 Zrotation Xrotation Yrotation

 JOINT Head

 {

 OFFSET 0.000000 4.630000 0.000000

 CHANNELS 3 Zrotation Xrotation Yrotation

 End Site

 {

 OFFSET 0.000000 3.419999 0.000000

 }

 }

 }

.....

 중간생략

.....

JOINT RightFoot

{

 OFFSET 0.000000 -17.360000 10.000000

 CHANNELS 3 Zrotation Xrotation Yrotation

 End Site

 {

 OFFSET 0.000000 -3.330000 0.000000

 }

 }

}

}

MOTION

Frames: 5

Frame Time: 0.033333

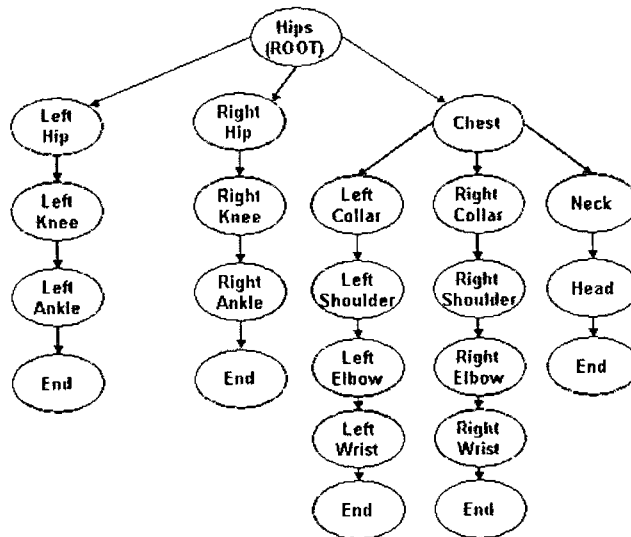
-122.986244 38.258511 -99.512604 4.360231 1.051365 49.898396 -4.219969

11.416894 -9.313691 10.370214 49.904026 -8.744551 0.756228 -55.780777 8.658085

-3.752158 0.000000 8.010931 27.170950 0.126464 22.705465 -64.299309 -57.68338

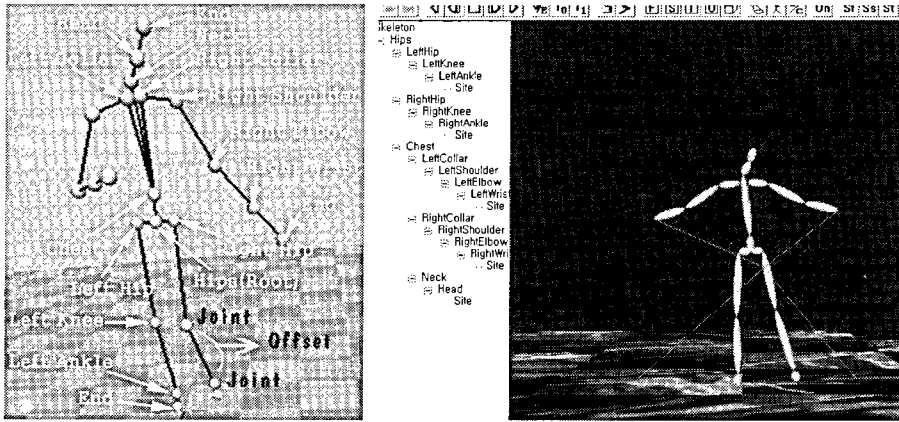
-90.615334	7.445413	5.542144	6.288157	19.604687	7.780634	-8.354941	-35.420570
-4.730030	51.332409	7.117136	-16.435846	12.885515	-17.076399	5.827448	-0.779622
1.450675	24.234940	0.294972	-4.224887	15.604630	-0.413905	-1.314118	12.794103
-4.443307	-1.780759	-31.080864	-2.041562	-0.811498	17.129311	-0.234632	2.840592
-9.367096	-0.643046						

상기(上記)된 BVH 모션 파일 내용을 간략히 분석해 보면 우선 Hierarchy 섹션에서 Root Hip은 계층구조(Hierarchy)상 최상위 부모(parent)로서 위치정보(OFFSET)가 영(0)으로 초기화 되는 것이고 그 하위 자식(Child) 구조들인 가슴(Chest)이나 목(Neck) 등은 부모와의 상대적 위치정보(OFFSET)에 의해 연결(JOINT)관계와 자유도(CHANNELS)를 갖게 되는 재귀 함수¹¹⁾(Recursive Function) 관계를 갖게 된다. 여기서 중요한 것은 부모와 자식 간의 상대적 X, Y, Z 위치 좌표 정보인 오프셋(OFFSET)인데 이 오프셋 정보에서 우리는 부모와 자식 간의 길이와 방향을 알 수 있기 때문에 쉽게 본을 드로잉할 수 있게 된다.



<그림 11> BVH 모션 파일에서 Hierarchy 섹션의 계층구조도

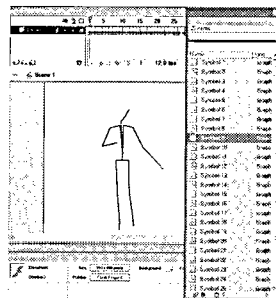
11) 자기 자신 함수의 호출을 재귀 호출(recursive call)이라 하고, 이러한 특징을 재귀(recursion)라 한다. 재귀를 다른 말로 되부름, 또는 순환이라 표현하며 재귀 호출을 허용하는 함수를 재귀 함수(recursive function)라 한다. 이 재귀 호출에서 주의할 점은 자신을 호출하면 새로 호출된 함수도 역시 자신을 호출하므로, 호출의 시슬을 종결하는 무엇인가가 없으면 재귀 호출은 끝없이 이어지게 돼서 무한 루프에 빠지게 된다. 즉, 재귀호출을 종료할 수 있게 그 무엇인가를 꼭 마련해 놓아야 한다. 그래서 <그림 11>의 Hierarchy 섹션의 계층구조의 최하위 노드에는 End가 설정되어 있는 것이다.



<그림 12> Hierarchy 섹션의 값으로 구현된 본(Bone)

다음으로 모션(Motion) 섹션을 분석해 보면 처음에 몇 프레임(Frames) 모션인지 알려주는 부분이 나오고 일반적인 샘플레이트(sample rate)인 초당 30프레임 애니메이션을 의미하는 Frame Time: 0.033333이라는 값을 보여준다. 그 다음의 수치 정보는 Hierarchy 섹션에서의 채널 값을 순서대로 나열한 것이다. 여기서 중요한 것은 최상위 부모의 채널 값은 6개지만 하위 자식의 채널 값은 3개라는 것과 회전 값의 순서가 X, Y, Z가 아닌 Z, X, Y라는 것이다. 물론 최상위 부모의 위치 정보 순서는 X, Y, Z 순이다. 순서가 중요한 것은 변환에 대한 처리는 행렬(Matrix)로 연산을 수행하게 되는데 행렬은 교환법칙이 성립되지 않기 때문이다.

BVH파일에 하이어아키(Hierarchy) 섹션을 참조하여 자동 생성된 본에는 이미 본(Bone) 간의 상대적 위치정보와 자유도를 트리구조로 가지고 있기 때문에 플래쉬와 같이 2D 이미지 객체를 라이브러리에 등록하여 재사용이 가능할 수 있게 본의 정보들을 따로 관리한 후에 2D의 3D화 응용 프로그램에서 제작한 2D 객체를 BVH파일에서 가져온 본의 정보에 링크(Link)를 걸어주면 바인딩이 이루어진다.



<그림 13> 플래쉬의 라이브러리에 등록된 2D화 된 본(Bone)

모션캡처된 데이터를 2D화된 본에 적용하려면, 그 전에 본의 모든 애니메이션 트리(Tree)들이 명백히 구조화되어 있어야 다중 XYZ 회전의 형태를 취하고 있는 예민한 동작들이 잘못된 관절에 적용되는 것을 막을 수 있다. 앞에서 언급한 모호 프로그램에서 모션 적용이 안 되는 결정적인 이유가 여기에 있다. <그림 10>의 파이프라인 방식으로 BVH 모션 파일의 하이어아키(Hierarchy) 섹션 정보를 이용하여 모션이 적용된 본을 제작하게 되면 위의 문제가 발생되지는 않지만 <그림 9>의 파이프라인 방식과 같이 여러 모션을 동일 캐릭터에 비순차적으로 적용할 수는 없게 된다. <그림 9>처럼 4단계 파이프라인을 적용하고자 할 경우에는 가능하면 본을 직접 만드는 방식보다는 구조화가 잘 이뤄진 본 들을 미리 템플릿으로 제작하여 그 본을 사용하는 방식으로 하면 된다.

2. 평가

사실 용어에 있어서 3D의 2D화라는 방식보다는 카툰렌더링 방식이 더욱 이해하기 쉽지만 굳이 3D의 2D화라고 사용하는 이유는 2D의 3D화와 구별하고자 함이다. 2D의 3D화 방식은 아직 제대로 된 프로그램이 없는 관계로 비록 상업화가 안 돼 잘 알려져 있지는 않지만 그나마 본에 모션을 적용시킬 수 있는 모션플레이쉬를 예로 들었고 3D는 맥스로, 3D의 2D화는 일러스트레이터를 각각 사용하여 동일한 시간(프레임)의 동일한 장면과 모델링을 P4 1.4GHz, 512 RAM의 노트북 시스템 사양에서 렌더링한 결과는 아래 <표1>과 같다. 렌더링 시간은 3D의 2D화 방식이 3D 방식보다 거의 10-20배에 가까운 엄청난 시간이 소요되었다. 물론 2D의 3D화 방식이 가장 빠르다. 그리고 221 프레임 약 7초정도의 애니메이션에서 렌더링된 파일 크기는 맥스로 렌더링 결과가 4메가이고 3D의 2D화는 12메가인데 비해서 2D의 3D화 애니메이션은 놀랍게도 0.2메가 밖에 안 된다는 것이다. 더욱 놀라운 점은 렌더링 할 장면의 폴리곤 수가 늘어남에 따라 그 차이는 더욱 커진다는 점이다. 크기에 있어서 2D의 3D화가 타 방식과 현격히 작은 이유는 앞에서 언급한 바와 같이 모션에 바인딩되는 이미지 즉 객체가 라이브러리에 등록되어 사용되기 때문이다. 그렇다면 기존의 3D의 2D화 방식은 라이브러리에 객체를 등록하지 않는다는 결론이 나온다. 뿐만 아니라 3D로 일단 처리 후에 2D로 변환 처리되기 때문에 시간의 소요가 크고 파일의 용량이 커지게 됨을 알 수 있다.

<표 1> 서로 다른 유형간의 성능 비교표

구분		3D	3D의 2D화	2D의 3D화
사용한 소프트웨어		3DS MAX 6.0	Illustrator 5.1	Motion Flash
렌더링 파일 형식		avi	swf	swf
렌더링 파일크기	폴리곤수 2,254개	525kb	3,259kb	131kb
	폴리곤수 15,798개	4,156kb	12,680kb	285kb
폴리곤수 7배 증가시 파일크기 증가율		8배	4배	2배
렌더링 소요 시간 (폴리곤수 15,798)		10분	2시간 20분	1분
공동 사항(샘플레이트/시간/화면크기)		30fps / 221 Frame(약 7초) / 320 X 240		
시스템 사양		Intel P4 Mobile CPU 1.4GHz, 512MB RAM, ATI Mobility RADEON 7500 16MB		

3D의 2D화는 렌더링 시에만 2D이기 때문에 예를 들어서 게임과 같이 소스를 이용하는 경우에 2D처리가 아닌 3D처리를 해야 하기 때문에 모바일이나 PDA게임과 같은 경우에는 크게 부담이 될 수 있다. 반면에 2D의 3D화는 원래의 소스부터가 2D이기 때문에 파일의 용량이 3D의 2D화보다 크게 작으며 실제 2D 작업된 파일보다도 더 작다. 그 이유는 실제 2D 작업에 들어가는 이미지가 애니메이션마다 개별적으로 존재해야 한다면 2D의 3D화는 한 개의 이미지가 라이브러리에 등록되어 있어서 전체 애니메이션 상에서 보면 실제 2D로 제작된 파일보다 크게 용량이 작음을 알 수 있다.

또한 기존의 2D 애니메이션 작업은 주로 키프레임과 트위닝 방식을 사용하기 때문에 애니메이션 키마다 이미지를 만들어야 됴므로 작업 시간이 오래 걸리게 되나 2D의 3D화 기법을 사용하게 되면 모션캡처 데이터를 활용하게 됴므로 처음에만 세팅을 하게 되면 애니메이션 작업 시간이 크게 줄 수 있게 된다. 동작의 자연스러움에 있어서 360도 회전 애니메이션 부분에서는 회전에 따른 이미지를 별도로 적용시켜줘야 하는 번거로움이 있지만 조금 신경을 써서 유의한다면 기존 2D보다 훨씬 자연스러운 동작을 연출해낼 수 있다.

IV. 결론 및 향후 연구

오늘날 애니메이션이나 게임 분야에서 마치 흑백 TV에서 컬러 TV로, 호출기에서 휴대전화기로 대체되어지는 것처럼 2D의 영역이 3D로 많은 부분이 축소되어 가고 있다. 하지만 2D와 3D는 서로 다른 장르일 뿐 2D가 3D의 하위 개념은 아니다. 오히려 서로 간의 많은

공통적인 부분의 차별화와 이질적인 부분의 통합화에 대한 연구 활동이 이루어지고 있다. 이러한 관점에서 3D의 2D화 방식으로 비실사 렌더링(Non-Photorealistic Rendering, NPR)이나 카툰렌더링(Cartoon Rendering) 등이 활발하게 연구 개발되어 실사적인 느낌의 3D는 그대로 추구하면서 반면에 3D의 차가운 느낌은 2D화하여 동화 같은 따뜻한 느낌으로 애니메이션이나 게임 분야에서 눈부신 발전을 보여 왔다. 반면에 2D의 3D화 방식은 학계나 업계에서 연구나 상업화가 거의 이루어지지 않고 있다.

3D 저작도구에만 존재했던 조명과 카메라 기능은 이미 많은 2D 저작도구에서 자연스럽게 활용되어지고 있다. 조명과 카메라 기법이 이제 더 이상 3D만의 고유한 전유물이 아니듯이 본과 모션 또한 2D에서 폭 넓게 구현되고 각종 애니메이션이나 게임에서 활용될 전망이다. 3D의 본과 모션이 2D에서 쉽게 구현된다면 기존의 2D애니메이션이나 2D 관련 멀티미디어 콘텐츠 제작 시간과 경비를 크게 줄일 수 있게 되고 이미 구축된 많은 양의 모션 DB가 3D에 국한되지 않고 2D에도 활용될 수 있기 때문에 이에 따른 애니메이션이나 게임 등의 문화 콘텐츠 산업이 폭넓고 활발하게 진행될 수 있게 된다.

본 논문에서는 2D 애니메이션 제작 시에 모션 및 본의 활용에 대한 방법에 대하여 연구하였다. 기존 3D의 2D화 작업 방식은 3D로 거의 모든 작업을 한 후에 렌더링 시에만 2D를 적용시키는 기법이어서 2D의 느낌을 제대로 살리기도 어려움이 있고 상대적으로 긴 렌더링 시간 또한 소스 파일을 이용해야 하는 모바일이나 PDA 게임 방식에 도입하기에는 어려움이 많았다. 본 논문은 2D의 3D화로 모션이 적용된 본의 2D화를 거쳐 2D의 이미지 객체와 2D화된 본을 라이브러리에 등록한 후에 서로 바인딩하는 프로세스를 통해 3D의 모션적용 파이프라인을 수정하여 2D애니메이션에서 본과 모션을 활용할 수 있게 2D의 모션적용 파이프라인을 제시하였다.

BVH 모션 파일에는 위치정보와 회전정보는 있으나 크기에 관한 정보가 없다. 모션데이터를 적용할 캐릭터가 캡콤사의 스트리트파이터 게임에 등장하는 달심처럼 공격 시에 팔 다리가 늘어나거나 줄어드는 특수한 경우를 제외하고는 캐릭터의 모션에 본의 크기 변화는 크게 의미가 없기 때문이다. 하지만 부득이하게 그런 애니메이션을 제작해야 할 경우라면 후처리 작업 시에 3D에서는 몰핑으로, 2D에서는 트윈닝으로 처리를 하는 것이 더욱 좋을 것이다. 크기 변환에 대한 문제보다는 향후 2D의 3D화에서 360도 회전변환에 따른 이미지 왜곡을 최대한 방지하기 위한 최적의 알고리즘의 개발에 대한 연구가 필요하다고 본다.

참고문헌

- 김대중, <셀 애니메이션 제작의 이론과 실제>(서울: 초록배, 1995).
- 김영관, <확장 동작 그래프를 이용한 새로운 동작 생성>(아주대학교 미디어학과 석사 논문, 2004).
- 김용순, 김영수, <3차원 캐릭터 애니메이션 기술동향>(정보과학회지 17권 2호, 1999).
- 이제희, <모션 캡처의 과거, 현재, 그리고 미래>(정보과학회지 21권 7호, 2003).
- 이인호, 박찬중, <모션캡처 기술의 현황과 응용분야>(한국멀티미디어학회지 3권 1호, 1999).
- 한국만화애니메이션학회, <만화애니메이션연구>(서울: 한울, 2003).
- Adam Lake, “Cartoon Rendering Using Texture Mapping and Programmable Vertex Shaders”, Game Programming Gems 2, Charles River Media, 2001; pp.444-449.
- Bruce Gooch, “A Non-Photorealistic Lighting Model for Automatic Technical Illustration”, In Proceedings of ACM SIGGRAPH 98, 1998.
- Carl S. Marshall, “Cartoon Rendering: Real-time Silhouette Edge Detection and Rendering”, Game Programming Gems 2, Charles River Media, 2001; pp.437-439.
- Carl Marshall, “Stylized Rendering Techniques for Scalable Real-Time 3D ACM Animation”, In Symposium of Non-Photorealistic Animation and Rendering 2000, 2000.
- Kerlow, Issac V. “The Art of 3-D Computer Animation and Imaging”, VanNostrand Reinhold. 1996.
- K. Perlin, “Real Time Responsive Animation with Personality”, IEEE Transactions on Visualization and Computer Graphics, 5-15, Vol.1, no.1, March 1995.
- Marco Tombesi, “3ds max Skin Exporter and Animation Toolkit”, Game Programming Gems 2, Charles River Media, 2001; pp.138-149.
- Mark Giambruno, “3D Graphics & Animation”, New Riders, 2002.
- Weber, Jason, “Constrained Inverse Kinematics”, Game Programming Gems 3, Charles River Media, 2002.
- Woodland, Ryan, “Filling the GapsAdvanced Animation Using Stitching and Skinning”, Game Programming Gems. Charles River Media, 2000.
- <http://web.mit.edu/comm-forum/papers/furniss.html>.
- <http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>.